Payment Types

Some payment methods require special request flows or completely new request types to work. These pages will describe every usecase for a specific payment method in addition to the basics outlined in the core API documentation.

Available Remarks

- Special remarks - 3-D Secure
- Special remarks - Recurring transactions credit card
- Special remarks - Alipay (via PAYONE Account Connect)
- Special remarks - Amazon Pay
- Special remarks - Barzahlen
- Special remarks - paydirekt
- Special remarks - PAYONE Secure Invoice
- Special remarks - Klarna Payments
- Special remarks - PayPal
- Special remarks - Unzer Payments
- Special remarks - Postfinance
- Special remarks - Ratepay
- Special remarks - PAYONE Direct Debit
- Special remarks - WeChat Pay
- Special remarks - Trustly
- Special remarks - Multibanco
- Special Remarks - iDEAL (Recurring Transactions)
- Special Remarks - Apple Pay
- Special remarks - Bancontact

Recurring Concepts

Genericpayment Requests

Some payment methods require certain request types which don't qualify as payment request, e.g. to initiate a session or calculate values which are then used in later payment requests. These types of requests are called `genericpayment` requests

```
request=genericpayment
```

add_paydata[action] Parameters

Many payment methods use specific parameters which only apply to their usecases. To keep the core API as clean as possible, we use the `add_paydata[action]` parameter to send these special parameters to the payment method. One example for PayPal ECS could be

```
add_paydata[action]=setexpresscheckout
```

Workorderid

Most of these requests will have to be linked to payment requests later in the process. To achieve this we need a unique identifier for one request chain. This is done using the parameter `workorderid`. You'll receive this parameter as a synchronous response parameter after your first request of the chain. Much like the `txid` in classic payment requests, the `workorderid` has to be sent during every request in the process leading up to the actual payment requests.

```
workorderid=WX1A372MN6M298PL
```

Putting It All Together

you can see all principles applied in this example:

Request

```
add_paydata[action]=setexpresscheckout
aid=12345 (your aid)
amount=2500
api_version=3.11
backurl=https://example.com/back
clearingtype=wlt
currency=EUR
encoding=UTF-8
errorurl=https://example.com/error
key=3adxxxxxxxxxxxxxxxxxxxxxxxx1cb9e (your md5-hashed portal key)
mid=12345 (your mid)
mode=test
portalid=123456 (your portal id)
request=genericpayment
successurl=https://example.com/success
wallettype=PPE
```

Response

```
status=REDIRECT
redirecturl=https://www.sandbox.paypal.com/webscr?
useraction=continue&cmd=_express-checkout&token=EC-xxxxxxxxxxxxxxxxxxx
add_paydata[token]=EC-xxxxxxxxxxxxxxxxxx
workorderid=WX1A372MN6M298PL
```

This is an initial request to trigger a PayPal Express Checkout session. you can see that in the request, we're using `genericpayment` as request type and also triggering the express Checkout by setting `add_paydata[action]` to `setexpresscheckout`. Then in the Response we're receiving the `workorderid`, which we'll use in subsequent requests of the Express Checkout flow.

The further flow of this Express Checkout is described in PayPal - Express Checkout (ECS).